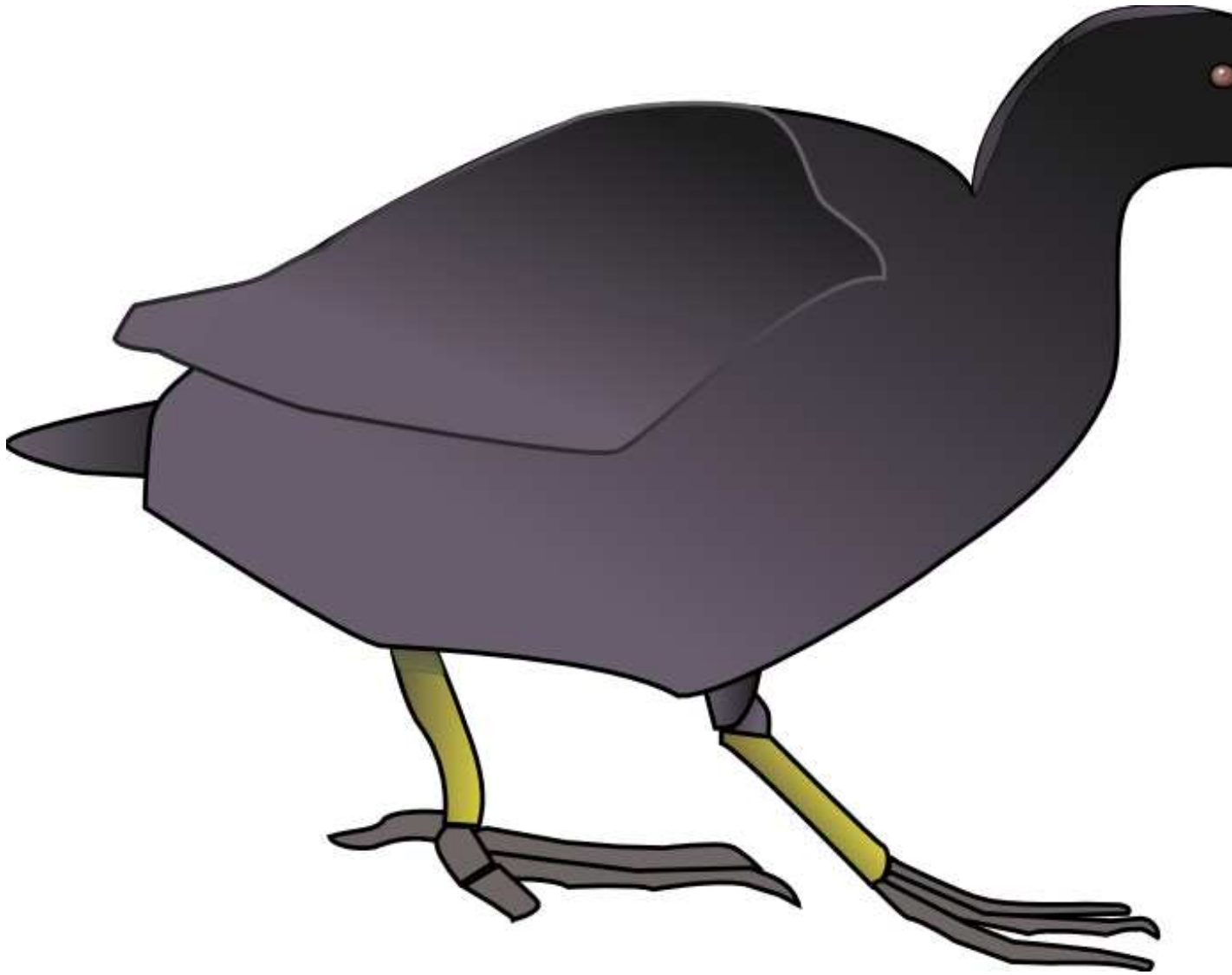


# The Coot User Manual

---

1st April 2004



---

Paul Emsley  
emsley@ysbl.york.ac.uk

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	This document . . . . .	7
1.2	What is Coot? . . . . .	7
1.3	What Coot is Not . . . . .	7
1.4	Hardware Requirements . . . . .	7
1.4.1	Mouse . . . . .	8
1.5	Environment Variables . . . . .	8
1.6	Command Line Arguments . . . . .	8
1.7	Web Page . . . . .	9
1.8	Crash . . . . .	9
<b>2</b>	<b>Mousing and Keyboarding</b>	<b>11</b>
2.1	Next Residue . . . . .	11
2.2	Keyboard Contouring . . . . .	11
2.3	Keyboard Rotation . . . . .	11
2.4	Keyboard Translation . . . . .	12
2.5	Keyboard Zoom and Clip . . . . .	12
2.6	Scrollwheel . . . . .	12
2.7	Selecting Atoms . . . . .	12
2.8	Virtual Trackball . . . . .	12
2.9	More on Zooming . . . . .	12
<b>3</b>	<b>General Features</b>	<b>13</b>
3.1	Version number . . . . .	13
3.2	Antialiasing . . . . .	13
3.3	Molecule Number . . . . .	13
3.4	Display Issues . . . . .	13
3.4.1	Origin Marker . . . . .	14
3.4.2	Raster3D output . . . . .	14
3.5	Display Manager . . . . .	14
3.6	The file selector . . . . .	15
3.6.1	File-name Filtering . . . . .	15
3.6.2	Filename Sorting . . . . .	15
3.7	Scripting . . . . .	15
3.7.1	Python . . . . .	15
3.7.2	Scheme . . . . .	16
3.7.3	State . . . . .	16
3.8	Backups and Undo . . . . .	16
3.8.1	Redo . . . . .	17
3.8.2	Restoring from Backup . . . . .	17
3.9	View Matrix . . . . .	17
3.10	Space Group . . . . .	17

3.11	Recentring View . . . . .	17
3.12	Clipping manipulation . . . . .	18
3.13	Background colour . . . . .	18
3.14	Unit Cell . . . . .	18
3.15	Rotation Centre Pointer . . . . .	18
3.15.1	Pointer Distances . . . . .	18
3.16	Crosshairs . . . . .	18
3.17	Frame Rate . . . . .	18
3.18	Program Output . . . . .	19
<b>4</b>	<b>Coordinate-Related Features</b>	<b>21</b>
4.1	Read coordinates . . . . .	21
4.1.1	Read multiple coordinate files . . . . .	21
4.2	Atom Info . . . . .	21
4.3	Atom Labeling . . . . .	21
4.4	Atom Colouring . . . . .	22
4.5	Bond Parameters . . . . .	22
4.5.1	Bond Thickness . . . . .	22
4.5.2	Display Hydrogens . . . . .	22
4.5.3	NCS Ghosts Coordinates . . . . .	22
4.5.4	NCS Maps . . . . .	23
4.6	Download coordinates . . . . .	23
4.7	Save Coordinates . . . . .	23
4.8	Anisotropic Atoms . . . . .	23
4.9	Symmetry . . . . .	23
4.10	Sequence View . . . . .	24
4.11	Environment Distances . . . . .	24
4.12	Distances and Angles . . . . .	24
4.13	Zero Occupancy Marker . . . . .	24
4.14	Mean, Median Temperature Factors . . . . .	24
4.15	Least-Squares Fitting . . . . .	24
4.16	More on Moving Molecules . . . . .	25
<b>5</b>	<b>Modelling and Building</b>	<b>27</b>
5.1	Regularization and Real Space Refinement . . . . .	27
5.1.1	Dictionary . . . . .	28
5.2	Rotate/Translate Zone . . . . .	28
5.3	Rigid Body Refinement . . . . .	28
5.4	Baton Build . . . . .	29
5.4.1	Building Backwards . . . . .	29
5.4.2	Undo . . . . .	30
5.4.3	Missing Skeleton . . . . .	30
5.5	C $\alpha$ $\rightarrow$ Mainchain . . . . .	30
5.6	Backbone Torsion Angles . . . . .	30
5.7	Rotamers . . . . .	30
5.7.1	Auto Fit Rotamer . . . . .	31
5.7.2	De-clashing residues . . . . .	31
5.8	Editing $\chi$ Angles . . . . .	32
5.8.1	Ligand Torsion angles . . . . .	32
5.9	Pep-flip . . . . .	32
5.10	Add Alternate Conformation . . . . .	32
5.11	Mutation . . . . .	32
5.11.1	Multiple mutations . . . . .	33
5.11.2	Mutate and Autofit . . . . .	33

5.11.3	Renumbering	33
5.12	Find Ligands	33
5.12.1	Flexible Ligands	34
5.12.2	Adding Ligands to Model	34
5.13	Find Waters	34
5.13.1	Blobs	35
5.13.2	Check Waters via Difference Map	35
5.14	Add Terminal Residue	35
5.15	Add OXT Atom to Residue	35
5.16	Add Atom at Pointer	36
5.17	Merge Molecules	36
5.18	Running Refmac	36
5.19	Clear Pending Picks	37
5.20	Delete	37
5.21	Sequence Assignment	37
5.22	Building Links and Loops	37
5.23	Setting Occupancies	37
<b>6</b>	<b>Map-Related Features</b>	<b>39</b>
6.1	Maps in General	39
6.2	Create a map	39
6.2.1	Reading CIF data	39
6.3	Map Contouring	39
6.4	Map contour “scrolling” limits	40
6.5	Map Line Width	40
6.6	“Dynamic” Map colouring	40
6.7	Difference Map Colouring	41
6.8	Map Sampling	41
6.9	Dragged Map	41
6.10	Dynamic Map Sampling and Display Size	41
6.11	Skeletonization	41
6.12	Masks	42
6.13	Trimming	42
<b>7</b>	<b>Validation</b>	<b>43</b>
7.1	Ramachandran Plots	43
7.2	Chiral Volumes	43
7.3	Blobs: a.k.a. Unmodelled density	43
7.4	Check Waters by Difference Map	44
7.5	Validation Graphs	44
7.5.1	Residue Density Fit	44
7.5.2	Rotamer Analysis	44
7.5.3	Temperature Factor Variance	44
7.5.4	Peptide $\omega$ Distortion	44
7.5.5	Geometry	44
<b>8</b>	<b>Hints</b>	<b>45</b>
8.1	Getting out of “Translate” Mode	45
8.2	Getting out of “Label Atom Only” Mode	45
8.3	Button Labels	45
8.4	Picking	45
8.5	Resizing View	45
8.6	Map	45
8.7	Slow Computer Configuration	46



# Chapter 1

## Introduction

### 1.1 This document

This document is the Coot User Manual, giving a brief overview of the interactive features. Other documentations includes (or it is planned to include) the Coot Reference Manual and the Coot Tutorial. These documents should be distributed with the source code.

### 1.2 What is Coot?

Coot is a stand-alone portion of CCP4's Molecular Graphics project. Its focus is crystallographic model-building and manipulation rather than representation (*i.e.* more like Frodo than Rasmol).

Coot is Free Software. You can give it away. If you don't like the way it behaves, you can fix it yourself.

### 1.3 What Coot is Not

Coot is not:

- CCP4's official Molecular Graphics program<sup>1</sup>
- a program to do refinement<sup>2</sup>
- a database, in any respect
- a protein crystallographic suite<sup>3</sup>.

### 1.4 Hardware Requirements

The code is designed to be portable to any Unix-like operating system<sup>4</sup>. Coot certainly runs on SGI IRIX64, RedHat Linux of various sorts, SuSe Linux<sup>5</sup> and MacOS X (10.2). The sgi Coot binaries should also work on IRIX.

---

<sup>1</sup>Coot is *part of* that project. The official program (which contains parts of Coot), ccp4mg is under the direct control of Liz Potterton and Stuart McNicholas.

<sup>2</sup>although it does have a local refinement algorithm it is no substitute for REFMAC (a wrapper for REFMAC is available).

<sup>3</sup>that's the job of the CCP4 Program Suite.

<sup>4</sup>including Cygwin, but currently coot is "unstable" on a Cygwin system.

<sup>5</sup>so far only 8.2 verified.

If you want to port to some other operating system, you are welcome<sup>6</sup>. Note that your task will be eased by using GNU GCC to compile the programs components.

### 1.4.1 Mouse

Coot works best with a 3-button mouse and works better if it has a scroll-wheel too (see Chapter 2 for more details)<sup>7</sup>.

## 1.5 Environment Variables

Coot responds to several command line arguments that modify its behaviour.

`COOT_STANDARD_RESIDUES` The filename of the pdb file containing the standard amino acid residues in “standard conformation”<sup>8</sup>

`COOT_SCHEME_DIR` The directory containing auxiliary scheme files

`COOT_REF_STRUCTS` The directory containing a set of high resolution pdb files used as reference structures to build backbone atoms from C $\alpha$  positions

`COOT_REFMAC_LIB_DIR` Refmac’s CIF directory containing the monomers and link descriptions. In the future this may simply be the same directory in which refmac looks to find the library dictionary.

`COOT_RESOURCES_DIR` The directory that contains the splash screen image and the GTK application resources.

`COOT_BACKUP_DIR` The directory to which backup are written (if it exists as a directory). If it is not, then backups are written to the current directory (the directory in which coot was started).

And of course extension language environment variables are used too:

`PYTHONPATH` (for python modules)

`GUILLE_LOAD_PATH` (for guile modules)

Normally, these environment variables will be set correctly in the coot setup script (which can be found in the `setup` directory in the binary distribution. See the web site (Section 1.7) for setup details.

## 1.6 Command Line Arguments

Rather than using the GUI to read in information, you can use the following command line arguments:

- `--script` to run a script on start up
- `--pdb` for pdb/coordinates file
- `--data` for mtz, phs or mmCIF data file
- `--map` for a (currently only CCP4) map

So, for example, one might use:

```
coot --pdb pre-refinement.pdb --pdb post-refinement.pdb
```

<sup>6</sup>it’s Free Software after all and I could give you a hand.

<sup>7</sup>I can get by with a one button Machintosh - but it’s not ideal.

<sup>8</sup>as it is known in Clipper.



## 1.7 Web Page

Coot has a web page:

<http://www.yysbl.york.ac.uk/-emsley/coot>

There you can read more about the CCP4 molecular graphics project in general and other projects which are important for coot<sup>9</sup>.

The web page also contains an example “setup” file which assigns the environment variables to change the behaviour of Coot.

## 1.8 Crash

Coot might crash on you - it shouldn't.

There are backup files in the directory `coot-backup`<sup>10</sup>. You can recover the session (until the last edit) by reading in the `pdb` file that you started with last time and then use **File** → **Recover Session**...

I would like to know about coot crashing<sup>11</sup> so that I can fix it as soon as possible. If you want your problem fixed, this involves some work on your part sadly.

First please make sure that you are using the most recent version of coot. I will often need to know as much as possible about what you did to cause the bug. If you can reproduce the bug and send me the files that are needed to cause it, I can almost certainly fix - it<sup>12</sup> especially if you use the debugger (`gdb`) and send a backtrace too<sup>13</sup>.

---

<sup>9</sup>coot has several influences and dependencies, but these will not be discussed here in the User Manual.

<sup>10</sup>`$COOT_BACKUP_DIR` is used in preference if set

<sup>11</sup>The map-reading problem (documented in Section 6.1) is already known.

<sup>12</sup>now there's a hostage to fortune.

<sup>13</sup>to do so, please send me the output of the following: `$ gdb `which coot` corefile` and then at the (`gdb`) prompt type: `where`, where *corefile* is the core dump file, `core` or `core.4536` or some such.



## Chapter 2

# Mousing and Keyboarding

How do we move around and select things?

Left-mouse Drag	Rotate view
Ctrl Left-Mouse Drag	Translates view
Shift Left-Mouse	Label Atom
Right-Mouse Drag	Zoom in and out
Shift Right-Mouse Drag	Rotate View around Screen Z axis
Middle-mouse	Centre on atom
Scroll-wheel Forward	Increase map contour level
Scroll-wheel Backward	Decrease map contour level

See also Chapter 8 for more help.

### 2.1 Next Residue

"Space"                Next Residue  
"Shift" "Space"    Previous Residue  
See also "Recentring View" (Section 3.11).

### 2.2 Keyboard Contouring

Use + or - on the keyboard if you don't have a scroll-wheel.

### 2.3 Keyboard Rotation

By popular request keyboard equivalents of rotations have been added<sup>1</sup>:

Q	Rotate + X Axis
W	Rotate - X Axis
E	Rotate + Y Axis
R	Rotate - Y Axis
T	Rotate + Z Axis
Y	Rotate - Z Axis
I	Continuous Y Axis Rotation

---

<sup>1</sup>particularly for those with PowerMates (the amount of rotation can be changed to 2° (from the default 1°) using (set-idle-function-rotate-angle 2.0)).

## 2.4 Keyboard Translation

Keypad 3    Push View (+Z translation)  
Keypad .    Pull View (-Z translation)

## 2.5 Keyboard Zoom and Clip

N    Zoom out  
M    Zoom in  
D    Slim clip  
F    Fatten clip

## 2.6 Scrollwheel

When there is no map, using the scroll-wheel has no effect. If there is exactly one map displayed, the scroll-wheel will change the contour level of that map. If there are two or more maps, the map for which the contour level is changed can be set by `HID → Scrollwheel → Attach scroll-wheel to which map?` and selecting a map number.

## 2.7 Selecting Atoms

Several Coot functions require the selecting of atoms to specify a residue range (for example: Regularize, Refine (Section 5.1) or Rigid Body Fit Zone (Section 5.3)). Select atoms with the Left-mouse. See also Picking (Section 8.4).

Use the scripting function `(quanta-buttons)` to make the mouse functions more like other molecular graphics programs to which you may be more accustomed<sup>2</sup>.

## 2.8 Virtual Trackball

You may not completely like the way the molecule is moved by the mouse movement<sup>3</sup>. To change this, try: `HID → Virtual Trackball → Flat`. To do this from the scripting interface: `(set-vt 1)`<sup>4</sup>.

If you *do* want screen-z rotation, you can either use Shift Right-Mouse Drag or set the Virtual Trackball to Spherical Surface mode and move the mouse along the bottom edget of the screen.

## 2.9 More on Zooming

The function `(quanta-like-zoom)` adds the ability to zoom the view using just Shift + Mouse movement<sup>5</sup>.

There is also a Zoom slider (`Draw → Zoom`) for those without a right-mouse button.

---

<sup>2</sup>See also 2.9

<sup>3</sup>Mouse movement in “Spherical Surface” mde generates a component of (often undesirable) screen z-rotation, particularly noticeable when the mouse is at the edge of the screen.

<sup>4</sup>`(set-vt 0)` to turn it back to “Spherical” mode.

<sup>5</sup>this is off by default because I find it annoying.

## Chapter 3

# General Features

The map-fitting and model-building tools can be accessed by using **Calculate** → **Model/Fit/Refine...** Many functions have tooltips<sup>1</sup> describing the particular features and are documented in Chapter 5.

### 3.1 Version number

The version number of Coot can be found at the top of the “About” window (**Help** → **About**).

There is also a script function to return the version of coot:

```
(coot-version)
```

### 3.2 Antialiasing

Antialiasing (for what it’s worth) can be enabled using:

```
(set-do-anti-aliasing 1)
```

The default is 0 (off).

### 3.3 Molecule Number

Coot is based on the concept of molecules. Maps and coordinates are different representations of molecules. The access to the molecule is *via* the *molecule number*. It is often important therefore to know the molecule number of a particular molecule.

Molecule numbers can be found by clicking on an atom in that molecule (if it has coordinates of course). The first number in brackets in the resulting text in the console is the molecule number. The molecule number can also be found in Display Control window (Section 3.5). It is also displayed on the left-hand side of the molecule name in the option menus of the “Save Coordinates” and “Go To Atom” windows.

### 3.4 Display Issues

The “graphics” window is drawn using OpenGL. It is considerably smoother when using a 3D accelerated X server.

---

<sup>1</sup>Put your mouse over a widget for a couple of seconds, if that widget has a tooltip, it will pop-up in a yellow box.

The view is orthographic (*i.e.* the back is the same size as the front). The default clipping is about right for viewing coordinate data, but is often a little too “thick” for viewing electron density. It is easily changed (see Section 3.12).

Depth-cueing is linear and fixed on. There is no anti-aliasing<sup>2</sup>.

The graphics window can be resized, but it has a minimum size of 400x400 pixels.

### 3.4.1 Origin Marker

A yellow box called the “origin marker” marks the origin. It can be removed using:

```
(set-show-origin-marker 0)
```

Its state can be queried like this:

```
(show-origin-marker-state)
```

which returns an number (an integer).

### 3.4.2 Raster3D output

Output suitable for use by Raster3D’s “render” can be generated using the scripting function

```
(raster3d file-name)
```

where *file-name* is such as “test.r3d”<sup>3</sup>.

There is a keyboard key to generate this file, run “render” and display the image: Function key F8.

You can also use the function

```
(render-image)
```

which will create a file *coot.r3d*, from which “render” produces *coot.png*.

This png file is displayed using ImageMagick’s display program (by default). Use something like:

```
(set! coot-png-display-program "gqview")
```

to change that to different display program (“gqview” in this case).

To change the widths of the bonds and density “lines” use (for example):

```
(set-raster3d-bond-thickness 0.1)
```

and

```
(set-raster3d-density-thickness 0.01)
```

To turn off the representations of the atoms (spheres):

```
(set-renderer-show-atoms 0)
```

## 3.5 Display Manager

This is also known as “Map and molecule (coordinates) display control”. Here you can select which maps and molecules you can see and how they are drawn<sup>4</sup>. The “Display” and “Active” are toggle buttons, either depressed (active) or undeprassed (inactive). The “Display” buttons control whether a molecule (or map) is drawn and the “Active” button controls if the molecule is clickable<sup>5</sup> (*i.e.* if the molecule’s atoms can be labeled).

By default, the path names of the files are not displayed in the Display Manager. To turn them on:

```
(set-show-paths-in-display-manager 1)
```

<sup>2</sup>coot is not the program for snazzy graphics - CCP4mg is the program for that sort of thing.

<sup>3</sup>Also povray will be supported in the future.

<sup>4</sup>to a limited extent.

<sup>5</sup>the substantial majority of the time you will want your the buttons to be both either depressed or undeprassed, rarely one but not the other.

If you pull across the horizontal scrollbar in a Molecule view, you will see the “Render as” menu. You can use this to change between normal “Bonds (Colour by Atom)”, “Bonds (Colour by Chain)” and “C $\alpha$ ” representation. There is also available “No Waters” and “C $\alpha$  + ligands” representations.

## 3.6 The file selector

### 3.6.1 File-name Filtering

The “Filter” button in the fileselection filters the filenames according to extension. For coordinates files the extensions are “.pdb” “.brk” “.mmCIF” and others. For data: “.mtz”, “.hkl”, “.phs”, “.cif” and for (CCP4) maps “.ext”, “.msk” and “.map”. If you want to add to the extensions, the following functions are available:

```
(add-coordinates-glob-extension extension)
(add-data-glob-extension extension)
(add-map-glob-extension extension)
(add-dictionary-glob-extension extension)
```

where *extension* is something like: “.mycif”.

### 3.6.2 Filename Sorting

If you like your files initially sorted by date (rather than lexicographically, which is the default use:

```
(set-sticky-sort-by-date)
```

## 3.7 Scripting

There is an compile-time option of adding a script interpreter. Currently the options are python and guile. Hopefully, in the near future you will be able to use both in the same executable, but that’s not available today.

Hundreds of commands are made available for use in scripting by using SWIG. These are currently not well documented but can be found in the Coot Reference Manual or the source code (`c-interface.h`).

Commands described throughout this manual (such as `(vt-surface 1)`) can be evaluated directly by Coot by using the “Scripting Window” (Calculate → Scripting...). Note that you type the commands in the lower entry widget and the command gets echoed (in red) and the return value and any output is displayed in the text widget above. The typed command should be terminated with a carriage return<sup>6</sup>. Files<sup>7</sup> can be evaluated (executed) using Calculate → Run Script... Note that in scheme (the usual scripting language of Coot), the parentheses are important.

### 3.7.1 Python

Coot has an (optional) embedded python interpreter. Thus the full power of python is available to you. Coot will look for an initialization script (`$HOME/.coot.py`) and will execute it if found. This file should contain python commands that set your personal preferences.

<sup>6</sup>which causes the evaluation of the command.

<sup>7</sup>such as the Coot state file (Section 3.7.3).

### Python Commands

The scripting functions described in this manual are formatted suitable for use with guile, *i.e.*:

```
(function arg1 arg2...)
```

If you are using Python instead: the format needs to be changed to:

```
function(arg1,arg2...)
```

Note that dashes in guile function names become underscores for python, so that (for example) `( raster-screen-shot )` becomes `raster_screen_shot ( )`.

### 3.7.2 Scheme

The scheme interpreter is made available by embedding guile. The initialization script used by this interpreter is `$HOME/.coot`. This file should contain scheme commands that set your personal preferences.

### 3.7.3 State

The “state” of coot is saved on Exit and written to a file called `0-coot.state.scm` (scheme) `0-coot.state.py` (python). This state file contains information about the screen centre, the clipping, colour map rotation size, the symmetry radius, and other molecule related parameters such as filename, column labels, coordinate filename *etc.*.

Use Calculate → Run Script... to use this file to re-create the loaded maps and models that you had when you finished using Coot<sup>8</sup> last time. A state file can be saved at any time using `(save-state)` which saves to file `0-coot.state.scm` or `(save-state-filename "thing.scm")` which saves to file `thing.scm`.

When Coot starts it can optionally run the commands in `0-coot.state.scm`. Use `(set-run-state-file-status i)` to change the behaviour: `i` is 0 to never run this state file at startup, `i` is 1 to get a dialog option (this is the default) and `i` is 2 to run the commands without question.

## 3.8 Backups and Undo

By default, each time a modification is made to a model, the old coordinates are written out<sup>9</sup>. The backups are kept in a backup directory and are tagged with the date and the history number (lower numbers are more ancient<sup>10</sup>). The “Undo” function discards the current molecule and loads itself from the most recent backup coordinates. Thus you do not have to remember to “Save Changes” - coot will do it for you<sup>11</sup>.

If you have made changes to more than one molecule, Coot will pop-up a dialog box in which you should set the “Undo Molecule” *i.e.* the molecule to which the Undo operations will apply. Further Undo operations will continue to apply to this molecule until there are none left. If another Undo is requested Coot checks to see if there are other molecules that can be undone, if there is exactly one, then that molecule becomes the “Undo Molecule”, if there are more than one, then another Undo selection dialog will be displayed.

<sup>8</sup>in that particular directory.

<sup>9</sup>this might be surprising since this could chew up a lot of disk space. However, disk space is cheap compared to losing you molecule.

<sup>10</sup>The coordinates are written in pdb format.

<sup>11</sup>unless you tell it not to, of course - use (*e.g.*) `(turn-off-backup 0)` to turn off the backup (for molecule 0 in this case).



### 3.8.1 Redo

The “undone” modifications can be re-done using this button. This is not available immediately after a modification<sup>12</sup>.

### 3.8.2 Restoring from Backup

There may be certain circumstances<sup>13</sup> in which you wish to restore from a backup but can’t get it by the “Undo” mechanism described above. In that case, start coot as normal and then open the (typically most recent) coordinates file in the directory `coot-backup` (or the directory pointed to the environment variable `COOT_BACKUP_DIR` if it was set) . This file should contain your most recent edits. In such a case, it is sensible for neatness purposes to immediately save the coordinates (probably to the current directory) so that you are not modifying a file in the backup directory.

See also Section 1.8.

## 3.9 View Matrix

It is sometimes useful to use this to orient the view and export this orientation to other programs. The orientation matrix of the view can be displayed (in the console) using:

```
(view-matrix)
```

## 3.10 Space Group

Occasionally you may want to know the space group of a particular molecule. Interactively (for maps) you can see it using the Map Properties button in the Molecule Display Control dialog.

There is a scripting interface function that returns the space group for a given molecule<sup>14</sup>:

```
(show-spacegroup imol)
```

## 3.11 Recentring View

Use Control + left-mouse to drag around the view

or

middle-mouse over an atom. In this case, you will often see “slide-recentring”, the graphics smoothly changes between the current centre and the newly selected centre.

or

Use Draw → Go To Atom... to select an atom using the keyboard. Note that you can subsequently use “Space” in the “graphics” window (OpenGL canvas) to re-centre on the next Cα.

If you don’t want smooth recentring (sliding) Draw → Smooth Recentring → Off. You can also use this dialog to speed it up a bit (by decreasing the number of steps instead of turning it off).

<sup>12</sup>It works like the “Forwards” buttons in a web browser - which is not available immediately after viewing a new page.

<sup>13</sup>for example, if coot crashes.

<sup>14</sup>if no space group has been assigned it returns ``No spacegroup for this molecule``

## 3.12 Clipping manipulation

The clipping planes (a.k.a. “slab”) can be adjusted using `Edit → Clipping` and adjusting the slider. There is only one parameter to change and it affects both the front and the back clipping planes<sup>15</sup>. The clipping can also be changed using keyboard “D” and “F”.

One can “push” and “pull” the view in the screen-Z direction using keypad 3 and keypad “.” (see Section 2.4).

## 3.13 Background colour

The background colour can be set either using a GUI dialog (`Edit → Background Colour`) or the function `(set-background-colour 0.00 0.00 0.00)`, where the arguments are 3 numbers between 0.0 and 1.0, which respectively represent the red, green and blue components of the background colour. The default is (0.0, 0.0, 0.0) (black).

## 3.14 Unit Cell

If coordinates have symmetry available then unit cells can be drawn for molecules (`Draw → Cell & Symmetry → Show Unit Cell?`).

The unit cell of maps can be drawn without needing to setup CCP4 first.

## 3.15 Rotation Centre Pointer

There is a pink pointer at the centre of the screen that marks the rotation centre. The size of the pointer can be changed using `Edit → Pink Pointer Size...` or using scripting commands: `(set-rotation-centre-size 0.3)`.

### 3.15.1 Pointer Distances

The Rotation Centre Pointer is sometimes called simply “Pointer”. One can find distances to the pointer from any active set of atoms using “Pointer Distances” (under Measures). If you move the Pointer (*e.g.* by centering on an atom) and want to update the distances to it, you have to toggle off and on the “Show Pointer Distances” on the Pointer Distances dialog.

## 3.16 Crosshairs

Crosshairs can be drawn at the centre of the screen, using either “c”<sup>16</sup> in graphics window or `Draw → Crosshairs...`. The ticks are at 1.54Å, 2.7Å and 3.8Å.

## 3.17 Frame Rate

Sometimes, you might ask yourself “how fast is the computer?”<sup>17</sup>. Using `Calculate → Frames/Sec` you can see how fast the molecule is rotating, giving an

---

<sup>15</sup>I find a clipping level of about 3.5 to 4 comfortable for viewing electron density maps - it is a little “thinner” than the default startup thickness.

<sup>16</sup>and “c” again to toggle them off.

<sup>17</sup>compared to some other one.

indication of graphics performance. It is often better to use a map that is more realistic and stop the picture whizzing round. The output is written to the console, you need to give it a few seconds to “settle down”. It is best not to have other widgets overlaying the GL canvas as you do this.

The contouring elapsed time<sup>18</sup> gives an indication of CPU performance.

## 3.18 Program Output

Due to its “in development” nature (at the moment), Coot produces a lot of “console”<sup>19</sup> output - much of it debugging or “informational”. This will go away in due course. You are advised to run Coot so that you can see the console and the graphics window at the same time, since feedback from atom clicking (for example) is often written there rather than displayed in the graphics window.

- Output that starts “ERROR...” is a programming problem (and ideally, you should never see it).
- Output that starts “WARNING...” means that something propably unintended happened due to the unexpected nature of your input or file(s).
- Output that starts “DEBUG...” has (obviously enough) been added to aid debugging. Most of them should have been cleaned up before release, but as Coot is constantly being developed, a few may slip through. Just ignore them.

---

<sup>18</sup>prompted by changing the contour level.

<sup>19</sup>*i.e.* the terminal in which you started Coot.



## Chapter 4

# Coordinate-Related Features

### 4.1 Read coordinates

The format of coordinates that can be read by coot is either PDB or mmCIF. To read coordinates, choose **File** → **Read Coordinates** from the menu-bar. Immediately after the coordinates have been read, the view is (by default) recentred to the centre of this new molecule and the molecule is displayed. To disable the recentring of the view on reading a coordinates file, use: `(recentre-on-read-pdb 0)`.

#### 4.1.1 Read multiple coordinate files

The reading multiple files using the GUI is not available (at the moment). However the following scripting functions are available:

```
(read-pdb-all)
```

which reads all the “\*.pdb” files in the current directory

```
(multi-read-pdb glob-pattern dir)
```

which reads all the files matching *glob-pattern* in directory *dir*. Typical usage of this might be:

```
(multi-read-pdb "a*.pdb" ".")
```

Alternatively you can specify the files to be opened on the command line when you start coot (see Section 1.6).

### 4.2 Atom Info

Information about a particular atom is displayed in the text console when you click using middle-mouse. Information for all the atoms in a residue is available using **Info** → **Residue Info...**

The temperature factors and occupancy of the atoms in a residue can be set by using **Edit** → **Residue Info...**

### 4.3 Atom Labeling

Use **Shift** + left-mouse to label atom. Do the same to toggle off the label. The font size is changeable using **Edit** → **Font Size...** The newly centred atom is labelled by default. To turn this off use:

```
(set-label-on-recentre-flag 0)
```

Some people prefer to have atom labels that are shorter, without the slashes and residue name:

```
(set-brief-atom-labels 1)
```

## 4.4 Atom Colouring

The atom colouring system in coot is unsophisticated. Typically, atoms are coloured by element: carbons are yellow, oxygens red, nitrogens blue, hydrogens white and everything else green (see Section 3.5 for colour by chain). However, it is useful to be able to distinguish different molecules by colour, so by default coot rotates the colour map of the atoms (*i.e.* changes the H value in the HSV<sup>1</sup> colour system). The amount of the rotation depends on the molecule number and a user-settable parameter:

```
(set-colour-map-rotation-on-read-pdb 30).
```

The default value is 31°.

Also one is able to select only the Carbon atoms to change colour in this manner: `(set-colour-map-rotation-on-read-pdb-c-only-flag 1)`.

## 4.5 Bond Parameters

The various bond parameters can be set using the GUI dialog `Draw → Bond Parameters` or *via* scripting functions.

### 4.5.1 Bond Thickness

The thickness (width) of bonds of individual molecules can be changed. This can be done via the `Bond Parameters` dialog or the scripting interface:

```
(set-bond-thickness thickness imol)
```

where `imol` is the molecule number. The default thickness is 3.0. The bond thickness also applies to the symmetry atoms of the molecule. There is no means to change the bond thickness of a residue selection within a molecule.

### 4.5.2 Display Hydrogens

Initially, hydrogens are displayed. They can be undisplayed using

```
(set-draw-hydrogens mol-no 0)2
```

where `mol-no` is the molecule number.

### 4.5.3 NCS Ghosts Coordinates

It is occasionally useful when analysing non-crystallographically related molecules to have “images” of the other related molecules appear matched onto the current coordinates. As you read in coordinates in Coot, they are checked for NCS relationships and clicking on “Show NCS Ghosts” → “Yes” → “Apply” will create “ghost” copies of them over the reference chain<sup>3</sup>.

---

<sup>1</sup>Hue Saturation Value (Intensity).

<sup>2</sup>they can be redisplayed using `(set-draw-hydrogens mol-no 1)`.

<sup>3</sup>the reference chain is the first chain of that type in the coordinates file.

### 4.5.4 NCS Maps

Coot can use the relative transformations of the NCS-related molecules in a coordinates molecule to transform maps. Use **Calculate** → **NCS Maps...** to do this (note the NCS maps only make sense in the region of the reference chain (see above). This will also create an NCS averaged map<sup>4</sup>.

## 4.6 Download coordinates

Coot provides the possibility to download coordinates from an OCA<sup>5</sup> (e.g. EBI) server<sup>6</sup> (**File** → **Get PDB Using Code...**). A popup entry box is displayed into which you can type a PDB accession code. Coot will then connect to the web server and transfer the file. Coot blocks as it does this (which is not ideal) but on a semi-decent internet connection, it's not too bad. The downloaded coordinates are saved into a directory called `.coot`.

It is also possible to download mmCIF data and generate a map. This currently requires a properly formatted database structure factors mmCIF file<sup>7</sup>.

## 4.7 Save Coordinates

On selecting from the menus **File** → **Save Coordinates...** you are first presented with a list of molecules which have coordinates. As well as the molecule number, there is the molecule name - very frequently the name of the file that was read in to generate the coordinates in Coot initially. However, this is only a *molecule* name and should not be confused with the filename to which the coordinates are saved. The coordinates *filename* can be selected using the **Select Filename...** button.

If your filename ends in `.cif`, `.mmCIF` or `.mmCIF` then an mmCIF file will be written (not a "PDB" file).

## 4.8 Anisotropic Atoms

By default anisotropic atom information is not represented<sup>8</sup>. To turn them on, use **Draw** → **Anisotropic Atoms** → **Show Anisotropic Atoms?** → **Yes**, or the command: `(set-show-aniso 1)`.

You cannot currently display thermal ellipsoids<sup>9</sup> for isotropic atoms.

## 4.9 Symmetry

Coordinates symmetry is "dynamic". Symmetry atoms can be labeled<sup>10</sup>. Every time you recentre, the symmetry gets updated. The information shown contains the atom information and the symmetry operation number and translations needed to generate the atom in that position.

---

<sup>4</sup>that also only makes sense in the region of the reference chain.

<sup>5</sup>OCA is "goose" in Spanish (and Italian).

<sup>6</sup>the default is the Weizmann Institute - which for reasons I won't go into here is currently much faster than the EBI server.

<sup>7</sup>which (currently) only a fraction are.

<sup>8</sup>using thermal ellipsoids

<sup>9</sup>in the case of isotropic atoms, ellipsoids are spherical, of course.

<sup>10</sup>symmetry labels are in pale blue and also provide the symmetry operator number and the translations along the a, b and c axes.

The symmetry can be represented as C $\alpha$ s. This along with representation of the molecule as C $\alpha$ s (Section 3.5) allow the production of a packing diagram.

## 4.10 Sequence View

The protein is represented by one letter codes and coloured according to secondary structure. These one letter codes are active - if you click on them, they will change the centre of the graphics window - in much the same way as clicking on a residue in the Ramachandran plot.

## 4.11 Environment Distances

Environment distances are turned on using Info  $\rightarrow$  Environment Distances. . . . Contacts to other residues are shown and to symmetry-related atoms if symmetry is being displayed. The contacts are coloured by atom type<sup>11</sup>.

## 4.12 Distances and Angles

The distance between atoms can be found using Info  $\rightarrow$  Distance<sup>12</sup>. The result is displayed graphically, and written to the console.

## 4.13 Zero Occupancy Marker

Atoms of zero occupancy are marked with a grey spot. To turn off these markers, use:

```
(set-draw-zero-occ-markers 0)
```

Use an argument of 1 to turn them on.

## 4.14 Mean, Median Temperature Factors

Coot can be used to calculate the mean (average) and median temperatures factors:

```
(average-temperature-factor imol)
```

```
(median-temperature-factor imol)
```

-1 is returned if there was a problem<sup>13</sup>.

## 4.15 Least-Squares Fitting

There is currently no GUI specified for this, the scripting interface is as follows:

```
(simple-lsq-match ref-start-resno ref-end-resno ref-chain-id
  imol-ref mov-start-resno mov-end-resno mov-chain-id imol-mov
  match-type)
```

where:

*ref-start-resno* is the starting residue number of the reference molecule

*ref-end-resno* is the last residue number of the reference molecule

*mov-start-resno* is the starting residue number of the moving molecule

---

<sup>11</sup>contacts not involving a carbon atom are yellow.

<sup>12</sup>Use Angle for an angle, of course.

<sup>13</sup>e.g. this molecule was a map or a closed molecule.



*mov-end-resno* is the last residue number of the moving molecule

*match-type* is one of 'CA', 'main', or 'all'.

e.g.: (simple-lsq-match 940 950 "A" 0 940 950 "A" 1 'main)

More sophisticated (match molecule number 1 chain "B" on to molecule number 0 chain "A"):

```
(define match1 (list 840 850 "A" 440 450 "B" 'all))
(define match2 (list 940 950 "A" 540 550 "B" 'main))
(clear-lsq-matches)
(set-match-element match1)
(set-match-element match2)
(lsq-match 0 1); match mol number 1 one mol number 0.
```

## 4.16 More on Moving Molecules

There are scripting functions available for this sort of thing:

(molecule-centre *imol*)

will tell you the molecule centre of the *imol*th molecule.

(translate-by *imol* x-shift y-shift z-shift)

will translate all the atoms in molecule *imol* by the given amount (in Ångströms).

(move-molecule-to-screen-centre *imol*)

will move the *imol*th molecule to the current centre of the screen (sometimes useful for imported ligands). Note that this moves the atoms of the molecule - not just the view of the molecule.



## Chapter 5

# Modelling and Building

The functions described in this chapter manipulate, extend or build molecules and can be found under Calculate → Model/Fit/Refine. . . .

### 5.1 Regularization and Real Space Refinement

If you have CCP4 installed, coot will read the geometry restraints for refmac and use them in fragment (zone) idealization - this is called “Regularization”. The geometrical restraints are, by default, bonds, angles, planes and non-bonded contacts. You can additionally use torsion restraints by Calculate → Model/Fit/Refine. . . → Refine/Regularize Control → Use Torsion Restraints.

“RS (Real Space) Refinement” (after Diamond, 1971<sup>1</sup>) in Coot is the use of the map in addition to geometry terms to improve the positions of the atoms. Select “Regularize” from the “Model/Fit/Refine” dialog and click on 2 atoms to define the zone (you can of course click on the same atom twice if you only want to regularize one residue). Coot then regularizes the residue range. At the end Coot displays the intermediate atoms in white and also displays a dialog, in which you can accept or reject this regularization. In the console are displayed the  $\chi^2$  values of the various geometrical restraints for the zone before and after the regularization. Usually the  $\chi^2$  values are considerably decreased - structure idealization such as this should drive the  $\chi^2$  values toward zero.

The use of “Refinement” is similar - with the addition of using a map. The map used to refine the structure is set by using the “Refine/Regularize Control” dialog. If you have read/created only one map into Coot, then that map will be used (there is no need to set it explicitly).

Use, for example, (set-matrix 20.0) <sup>2</sup> to change the weight of the map gradients to geometric gradients. The higher the number the more weight that is given to the map terms<sup>3</sup>. The default is 150.0. This will be needed for maps generated from data not on (or close to) the absolute scale or maps that have been scaled (for example so that the sigma level has been scaled to 1.0).

For both “Regularize Zone” and “Refine Zone” one is able to use a single click to refine a residue range. Pressing “A” on the keyboard while selecting an atom in a residue will automatically create a residue range with that residue in the middle. By default the zone is extended one residue either side of the central residue. This can be changed to 2 either side using (set-refine-auto-range-step 2).

---

<sup>1</sup>Diamond, R. (1971). A Real-Space Refinement Procedure for Proteins. *Acta Crystallographica* **A27**, 436-452.

<sup>2</sup>set\_matrix(20.0) (using python).

<sup>3</sup>but the resulting  $\chi^2$  values are higher.

This is a useful feature

Intermediate (white) atoms can be moved around with the mouse (click and drag with left-mouse, by default). Refinement will proceed from the new atom positions when the mouse button is released. It is possible to create incorrect atom nomenclature and/or chiral volumes in this manner - so some care must be taken. Press the “A” key as you left-mouse click to move atoms more “locally” (rather than a linear shear) and Cntrl key as you left-mouse click to move just one atom.

To prevent the unintentional refinement of a large number of residues, there is a “heuristic fencepost” of 20 residues. A selection of than 20 residues will not be regularized or refined. The limit can be changed using the scripting function: *e.g.* `(set-refine-max-residues 30)`.

### 5.1.1 Dictionary

By default, the geometry dictionary entries for only the standard residues are read in at the start <sup>4</sup>. It may be that you particular ligand is not amongst these. To interactively add a dictionary entry use File → Import CIF Dictionary. Alternatively, you can use the function:

```
(read-cif-dictionary filename)
```

and add this to your `.coot` file (this may be the preferred method if you want to read the file on more than one occassion).

Note: the dictionary also provides the description of the ligand’s torsions.

## 5.2 Rotate/Translate Zone

“Rotate/Translate Zone” from the “Model/Fit/Refine” menu allows manual movement of a zone. After pressing the “Rotate/Translate Zone” button, select two atoms in the graphics canvas to define a residue range<sup>5</sup>, the second atom that you click will be the local rotation centre for the zone. The atoms selected in the moving fragment have the same alternate conformation code as the first atom you click. To actuate a transformation, click and drag horizontally across the relevant button in the newly-created “Rotation & Translation” dialog. The axis system of the rotations and translations are the screen coordinates. Alternatively <sup>6</sup>, you can click using left-mouse on an atom in the fragment and drag the fragment around. Use Control Left-mouse to move just one atom, rather than the whole fragment. Click “OK” when the transformation is complete.

## 5.3 Rigid Body Refinement

“Rigid Body Fit Zone” from the “Model/Fit/Refine” dialog provides rigid body refinement. The selection is zone-based<sup>7</sup>. So to refine just one residue, click on one atom twice.

Sometimes no results are displayed after Rigid Body Fit Zone. This is because the final model positions had too many final atom positions in negative density. If you want to over-rule the default fraction of atoms in the zone that have an acceptable fit (0.75), to be (say) 0.25:

```
(set-rigid-body-fit-acceptable-fit-fraction 0.25)
```

<sup>4</sup>And a few extras, such as phosphate

<sup>5</sup>if you want to move only one residue, then click the same atom twice.

<sup>6</sup>like Refinement and Regularization

<sup>7</sup>like Regularization and Refinement.

## 5.4 Baton Build

Baton build is most useful if a skeleton is already calculated and displayed (see Section 6.11). When three or more atoms have been built in a chain, Coot will use a prior probability distribution for the next position based on the position of the previous three. The analysis is similar to Oldfield & Hubbard<sup>8</sup>, however it is based on a more recent and considerably larger database.

Little crosses are drawn representing directions in which it is possible that the chain goes, and a baton is drawn from the current point to one of these new positions. If you don't like this particular direction<sup>9</sup>, use Try Another. The list of directions is scored according to the above criterion and sorted so that the most likely is at the top of the list and displayed first as the baton direction.

When starting baton building, be sure to be about 3.8Å from the position of the first-placed C $\alpha$ , this is because the next C $\alpha$  is placed at the end of the baton, the baton root being at the centre of the screen. So, when trying to baton-build a chain starting at residue 1, centre the screen at about the position of residue 2.

Occasionally, every point is not where you want to position the next atom. In that case you can either shorten or lengthen the baton, or position it yourself using the mouse. Use "b" on the keyboard to swap to baton mode for the mouse<sup>10</sup>.

Baton-built atoms are placed into a molecule called "Baton Atom" and it is often sensible to save the coordinates of this molecule before quitting coot.

If you try to trace a high resolution map (1.5Å or better) you will need to increase the skeleton search depth from the default (10), for example:

```
(set-max-skeleton-search-depth 20)
```

Alternatively, you could generate a new map using data to a more moderate resolution (2Å), the map may be easier to interpret at that resolution anyhow<sup>11</sup>.

The guide positions are updated every time the "Accept" button is clicked. The molecule name for these atoms is "Baton Build Guide Points" and it is not usually necessary to keep them.

### 5.4.1 Building Backwards

The following scenario is not uncommon: you find a nice stretch of density and start baton building in it. After a while you come to a point where you stop (dismissing the baton build dialog). You want to go back to where you started and build the other way. How do you do that?

- Use the command: `(set-baton-build-params start-resno chain-id "backwards")`, where `start-resno` would typically be 0<sup>12</sup> and `chain-id` would be "" (default).
- Recentre the graphics window on the first atom of the just-build fragment
- Select "C $\alpha$  Baton Mode" and select a baton direction that goes in the "opposite" direction to what is typically residue 2. This is slightly awkward because the initial baton atoms build in the "opposite" direction are not dependent on the first few atoms of the previously build fragment.

<sup>8</sup>T. J. Oldfield & R. E. Hubbard. "Analysis of C-Alpha Geometry in Protein Structures" *Proteins-Structure Function and Genetics* **18**(4) 324 – 337.

<sup>9</sup>which is quite likely at first since coot has no knowledge of where the chain has been and cannot score according to geometric criteria.

<sup>10</sup>"b" again toggles the mode off.

<sup>11</sup>high-resolution map interpretation is planned.

<sup>12</sup>i.e. one less than the starting residue in the forward direction (defaults to 1).

### 5.4.2 Undo

There is also an “Undo” button for baton-building. Pressing this will delete the most recently placed C $\alpha$  and the guide points will be recalculated for the previous position. The number of “Undo”s is unlimited. Note that you should use the “Undo” button in the Baton Build dialog, not the one in the “Model/Fit/Refine” dialog (Section 3.8).

### 5.4.3 Missing Skeleton

Sometimes (especially at loops) you can see the direction in which the chain should go, but there is no skeleton (see Section 6.11) is displayed (and consequently no guide points) in that direction. In that case, “Undo” the previous atom and decrease the skeletonization level (Edit  $\rightarrow$  Skeleton Parameters  $\rightarrow$  Skeletonization Level). Accept the atom (in the same place as last time) and now when the new guide points are displayed, there should be an option to build in a new direction.

## 5.5 C $\alpha$ $\rightarrow$ Mainchain

Mainchain can be generated using a set of C $\alpha$ s as guide-points (such as those from Baton-building) along the line of Esnouf<sup>13</sup> or Jones and coworkers<sup>14</sup>. Briefly, 6-residue fragments of are generated from a list of high-quality<sup>15</sup> structures. The C $\alpha$  atoms of these fragments are matched against overlapping sets of the guide-point C $\alpha$ s. The resulting matches are merged to provide positions for the mainchain (and C $\beta$ ) atoms. This procedure works well for helices and strands, but less well<sup>16</sup> for less common structural features.

This function is also available from the scripting interface:

```
(db-mainchain imol chain-id resno-start resno-end direction)
where direction is either "backwards" or "forwards".
```

## 5.6 Backbone Torsion Angles

It is possible to edit the backbone  $\phi$  and  $\psi$  angles indirectly using an option in the Model/Fit/Refine’s dialog: “Edit Backbone Torsions..”. When clicked and an atom of a peptide is selected, this produces a new dialog that offers “Rotate Peptide” which changes this residues  $\psi$  and “Rotate Carbonyl” which changes  $\phi$ . Click and drag across the button<sup>17</sup> to rotate the moving atoms in the graphics window. You should know, of course, that making these modifications alter the  $\phi/\psi$  angles of more than one residue.

## 5.7 Rotamers

The rotamers are generated from the backbone independent sidechain library of Roland Dunbrack and co-workers<sup>18</sup>. According to this analysis, some sidechains

<sup>13</sup>R. M. Esnouf “Polyalanine Reconstruction from C $\alpha$  Positions Using the Program CALPHA Can Aid Initial Phasing of Data by Molecular Replacement Procedures” *Acta Cryst.*, D53, 666-672 (1997).

<sup>14</sup>T.A. Jones & S. Thirup “Using known substructures in protein model building and crystallography” *EMBO J.* 5, 819-822 (1986).

<sup>15</sup>and high resolution

<sup>16</sup>*i.e.* there are severely misplaced atoms

<sup>17</sup>as for Rotate/Translate Zone (Section 5.2).

<sup>18</sup>R. L. Dunbrack, Jr. & F. E. Cohen. “Bayesian statistical analysis of protein sidechain rotamer preferences” *Protein Science*, 6, 1661-1681 (1997).

have many rotamer options<sup>19</sup>. By default only rotamers with a probability (as derived from the structural database) greater than 1% are considered. The initial position is the most likely for that residue type<sup>20</sup>.

Use keyboard "." and "," to cycle round the rotamers.

### 5.7.1 Auto Fit Rotamer

"Auto Fit Rotamer" will try to fit the rotamer to the electron density. Each rotamer is generated, rigid body refined and scored according to the fit to the map. Fitting the second conformation of a dual conformation in this way will often fail - the algorithm will pick the best fit to the density - ignoring the position of the other atoms.

The algorithm doesn't know if the other atoms in the structure are in sensible positions. If they are, then it is sensible not to put this residue too close to them, if they are not then there should be no restriction from the other atoms as to the position of this residue - the default is "are sensible", which means that the algorithm is prevented from finding solutions that are too close to the atoms of other residues. (`set-rotamer-check-clashes 0`) will stop this.

There is a scripting interface to auto-fitting rotamers:

```
(auto-fit-best-rotamer resno alt-loc ins-code chain-id
  imol-coords imol-map clash-flag lowest-rotamer-probability)
```

where:

*resno* is the residue number

*alt-loc* is the alternate/alternative location symbol (e.g. "A" or "B", but most often " ")

*ins-code* is the insertion code (usually " ")

*imol-coords* is the molecule number of the coordinates molecule

*imol-map* is the molecule number of the map to which you wish to fit the side chains

*clash-flag* should the positions of other residues be included in the scoring of the rotamers (i.e. clashing with other other atoms gets marked as bad/unlikely)

*lowest-rotamer-probability*: some rotamers of some side chains are so unlikely that they shouldn't be considered - typically 0.01 (1%).

### 5.7.2 De-clashing residues

Sometimes you don't have a map<sup>21</sup> but nevertheless there are clashing residues<sup>22</sup> (for example after mutation of a residue range) and you need to rotate side-chains to a non-clashing rotamer. There is a scripting interface:

```
(de-clash imol chain-id start-resno end-resno)
```

*start-resno* is the residue number of the first residue you wish to de-clash.

*end-resno* is the residue number of the last residue you wish to de-clash

*imol* is the molecule number of the coordinates molecule

This interface will not change residues with insertion codes or alternate conformation. The *lowest-rotamer-probability* is set to 0.01.

<sup>19</sup>LYS, for example has 81.

<sup>20</sup>Use e.g. (`set-rotamer-lowest-probability 0.5`) to change the probability lower limit for the rotamer selection (note that this is a percentage, therefore 0.5(%) is quite low and will allow the choice of more rotamers than the default).

<sup>21</sup>for example, in preparation of a model for molecular replacement

<sup>22</sup>atoms of residues that are too close to each other

## 5.8 Editing $\chi$ Angles

Instead of using Rotamers, one can instead change the  $\chi$  angles (often called “torsions”) “by hand” (using “Edit Chi Angles” from the “Model/Fit/Refine” dialog). To edit a residue’s  $\chi_1$  press “1”: to edit  $\chi_2$ , “2”:  $\chi_3$  “3” and  $\chi_4$  “4”. Use left-mouse click and drag to change the  $\chi$  value. Use keyboard “0”<sup>23</sup> to go back to ordinary view mode at any time during the editing. Alternatively, one can use the “View Rotation Mode” or use the Ctrl key when moving the mouse in the graphics window. Use the Accept/Reject dialog when you have finished editing the  $\chi$  angles.

It should be emphasised that for standard residues this is an option of last resort - use the other rotamer manipulation options first.

### 5.8.1 Ligand Torsion angles

For ligands, you will need to read the mmCIF file that contains a description of the ligand’s geometry (see Section 5.1.1). By default, torsions that move hydrogens are not included. Only 9 torsion angles are available from the keyboard torsion angle selection.

## 5.9 Pep-flip

Coot uses the same pepflip scheme as is used in O (*i.e.* the C, N and O atoms are rotated 180° round a line joining the C $\alpha$  atoms of the residues involved in the peptide). Flip the peptide again to return the atoms to their previous position.

## 5.10 Add Alternate Conformation

This allows the addition of alternate (dual, triple *etc.*) conformations to the picked residue. By default, this provides a choice of rotamer (Section 5.7). If there are not the correct main chain atoms a rotamer choice cannot be provided, and Coot falls back to providing intermediate atoms.

The default occupancy for new atoms is 0.5. This can be changed by using the slider on the rotamer selection window or by using the scripting function:

```
(set-add-alt-conf-new-atoms-occupancy 0.4)
```

## 5.11 Mutation

Mutations are available on a 1-by-1 basis using the graphics. After selecting “Mutate...” from the “Model/Fit/Refine” dialog, click on an atom in the graphics. A “Residue Type” window will now appear. Select the new residue type you wish and the residue in the graphics is updated to the new residue type<sup>24</sup>. The initial position of the new rotamer is the *a priori* most likely rotamer. Note that in interactive mode, such as this, a residue type match<sup>25</sup> will not stop the mutation action occurring.

<sup>23</sup>that’s “zero”.

<sup>24</sup>Note that selecting a residue type that matches the residue in the graphics will also result in a mutation

<sup>25</sup>*i.e.* the current residue type matches the residue type to which you wish to mutate the residue



### 5.11.1 Multiple mutations

This dialog can be found under Calculate → Mutate Residue Range. A residue range can be assigned a sequence and optionally fitted to the map. This is useful converting a poly-ALA model to the correct sequence<sup>26</sup>.

Multiple mutations are also supported *via* the scripting interface. Unlike the single residue mutation function, a residue type match *will* prevent a modification of the residue<sup>27</sup>. Two functions are provided: To mutate a whole chain, use (mutate-chain *imol chain-id sequence*) where:

*chain-id* is the chain identifier of the chain that you wish to mutate (e.g. "A") and

*imol* is molecule number.

*sequence* is a list of single-letter residue codes, such as "GYRESDF" (this should be a straight string with no additional spaces or carriage returns).

Note that the number of residues in the sequence chain and those in the chain of the protein must match exactly (*i.e.* the whole of the chain is mutated (except residues that have a matching residue type).)

To mutate a residue range, use

```
(mutate-residue-range chain-id start-res-no stop-res-no
sequence)
```

where

*start-res-no* is the starting residue for mutation

*stop-res-no* is the last residue for mutation, *i.e.* using values of 2 and 3 for *start-res-no* and *stop-res-no* respectively will mutate 2 residues.

Again, the length of the sequence must correspond to the residue range length.

### 5.11.2 Mutate and Autofit

The function combines Mutation and Auto Fit Rotamer and is the easiest way to make a mutation and then fit to the map.

### 5.11.3 Renumbering

Renumbering is straightforward using the renumber dialog available under Calculate → Renumber Residue Range. . . . There is also a scripting interface:

```
(renumber-residue-range imol chain-id start-res-no last-resno
offset)
```

## 5.12 Find Ligands

You are offered a selection of maps to search (you can only choose one at a time) and a selection of molecules that act as a mask to this map. Finally you must choose which ligand types you are going to search for in this map<sup>28</sup>. Only molecules with less than 400 atoms are suggested as potential ligands. New ligands are placed where the map density is and protein (mask) atoms are *not*. The masked map is searched for clusters using a default cut-off of 1.0 $\sigma$ . In weak density this cut-off may be too high and in such a case the cut-off value can be changed using something such as:

```
(set-ligand-cluster-sigma-level 0.8)
```

<sup>26</sup>e.g. after using Ca → Mainchain.

<sup>27</sup>*i.e.* the residue atoms will remain untouched

<sup>28</sup>you can search for many different ligand types.

However, if the map to be searched for ligands is a difference map, a cluster level of 2.0 or 3.0 would probably be more appropriate<sup>29</sup>.

Each ligand is fitted with rigid body refinement to each potential ligand site in the map and the best one for each site selected and written out as a pdb file. The clusters are sorted by size, the biggest one first (with an index of 0). The output placed ligands files have a prefix “best-overall” and are tagged by the cluster index and residue type of the best fit ligand in that site.

By default, the top 10 sites are tested for ligands - to increase this use:

```
(set-ligand-n-top-ligands 20)
```

### 5.12.1 Flexible Ligands

If the “Flexible?” checkbox is activated, coot will generate a number of variable conformations (default 100) by rotating around the rotatable bonds (torsions). Each of these conformations will be fitted to each of the potential ligand sites in the map and the best one will be selected (again, if it passes the fitting criteria above).

Before you search for flexible ligands you must have read the mmCIF dictionary for that particular ligand residue type (File → Import CIF dictionary).

Use:

```
(set-ligand-flexible-ligand-n-samples n-samples)
```

where *n-samples* is the number of samples of flexibility made for each ligand. The more the number of rotatable bonds, the bigger this number should be.

### 5.12.2 Adding Ligands to Model

After successful ligand searching, one may well want to add that displayed ligand to the current model (the coordinates set that provided the map mask). To do so, use Merge Molecules (Section 5.17).

## 5.13 Find Waters

As with finding ligands, you are given a choice of maps, protein (masking) atoms. A final selection has to be made for the cut-off level, note that this value is the number of standard deviation of the density of the map *after* the map has been masked. Then the map is masked by the masking atoms and a search is made of features in the map about the electron density cut-off value. Waters are added if the feature is approximately water-sized and can make sensible hydrogen bonds to the protein atoms. The new waters are optionally created in a new molecule called “Waters”.

You have control over several parameters used in the water finding:

```
(set-write-peaksearched-waters)
```

which writes `ligand-waters-peaksearch-results.pdb`, which contains the water peaks (from the clusters) without any filtering and `ligand-waters.pdb` which are a disk copy filtered waters that have been either added to the molecule or from which a new molecule has been created.

```
(set-ligand-water-spherical-variance-limit min-d max-d)
```

sets the minimum and maximum allowable distances between new waters and the masking molecule (usually the protein).

```
(set-ligand-water-spherical-variance-limit varlim)
```

sets the upper limit for the density variance around water atoms. The default is 0.12.

The map that is made by the protein and is searched to find the waters is written out in CCP4 format as “`masked-for-waters.map`”.

<sup>29</sup>less likely to generate spurious sites.

### 5.13.1 Blobs

After a water search, Coot will create a blobs dialog (see Section 7.3).

### 5.13.2 Check Waters via Difference Map

Another check of waters that one can perform is the following:

```
(check-waters-by-difference-map imol-coords imol-diff-map)
```

where *imol-coords* is the molecule number of the coordinates that contain the waters to be checked

*imol-diff-map* is the molecule number of the difference map (it must be a difference map, not an “ordinary” map). This difference map must have been calculated using the waters. So there is no point in doing this check immediately after “Find Waters”. You will need to run Refmac first<sup>30</sup>.

This analysis will return a list of water atoms that have outstandingly high local variance of the difference map (by default a sphere of 1.5Å centred about the atom position). This analysis might find waters that are actually something else, for example: part of a ligand, a sulfate, an anion or cation, only partially occupied or should be deleted entirely. Coot<sup>31</sup> doesn’t decide what should be done about these atoms, it merely brings them to your attention. It may be interesting to use an anomalous map to do this analysis.

There is no GUI for this feature.

## 5.14 Add Terminal Residue

This creates a new residue at the C or N terminus by fitting to the map.  $\phi/\psi$  angle pairs are selected at random based on the Ramachandran plot probability (for a generic residue). By default there are 100 trials. It is possible that a wrong position will be selected for the terminal residue and if so, you can reject this fit and try again with Fit Terminal Residue<sup>32</sup>. Each of the trial positions are scored according to their fit to the map<sup>33</sup> and the best one selected. It is probably a good idea to run “Refine Zone” on these new residues.

(`set-terminal-residue-do-rigid-body-refine 0`) will disable rigid body fitting of the terminal residue fragment for each trial residue position (the default is 1 (on)) - this may help if the search does not provide good results.

(`set-add-terminal-residue-n-phi-psi-trials 50`) will change the number of trials (default is 100).

## 5.15 Add OXT Atom to Residue

At the C-terminus of a chain of amino-acid residues, there is a “modification” so that the C-O becomes a carbonyl, *i.e.* an extra (terminal) oxygen (OXT) needs to be added. This atom is added so that it is in the plane of the C $\alpha$ , C and O atoms of the residue.

Scripting usage:

```
(add-OXT-to-residue imol residue-number
  insertion-code chain-id)34,
  where insertion-code is typically "".
```

<sup>30</sup>and remember to check the difference map button in the “Run Refmac” dialog

<sup>31</sup>as yet

<sup>32</sup>usually if this still fails after two repetitions then it never seems to work.

<sup>33</sup>The map is selected using “Refine/Regularize Control”

<sup>34</sup>*e.g.* (`add-OXT-to-residue 0 428 "" "A"`)

Note, in order to place OXT, the N, CA, C and O atoms must be present in the residue - if (for example) the existing carbonyl oxygen atom is called "OE1" then this function will not work.

## 5.16 Add Atom at Pointer

By default, "Add Atom At Pointer" will pop-up a dialog from which you can choose the atom type you wish to insert<sup>35</sup>. Using `(set-pointer-atom-is-dummy 1)` you can by-pass this dialog and immediately create a dummy atom at the pointer position. Use an argument of 0 to revert to using the atom type selection pop-up on a button press.

The atoms are added to a new molecule called "Pointer Atoms". They should be saved and merged with your coordinates outside of Coot.

## 5.17 Merge Molecules

This dialog can be found under "Calculate" in the main menubar. This is typically used to add molecule fragments or residues that are in one molecule to the "working" coordinates<sup>36</sup>.

## 5.18 Running Refmac

Use the "Run Refmac..." button to select the dataset and the coordinates on which you would like to run Refmac. Note that only dataset which had Refmac parameters set as the MTZ file was read are offered as dataset that can be used with Refmac. By default, Coot displays the new coordinates and the new map generated from refmac's output MTZ file. Optionally, you can also display the difference map.

You can add extra parameters (data lines) to refmac's input by storing them in a file called `refmac-extra-params` in the directory in which you started coot.

Coot "blocks"<sup>37</sup> until Refmac has terminated<sup>38</sup>.

The default refmac executable is `refmac5` it is presumed to be in the path. If you don't want this, it can be overridden using a re-definition either at the scripting interface or in one's `.coot` file *e.g.*:

```
(define refmac-exec "/e/refmac-new/bin/refmac5.6.3")
```

After running refmac several times, you may find that you prefer if the new map that refmac creates (after refmac refinement) is the same colour as the previous one (from before this refmac refinement). If so, use:

```
(set-keep-map-colour-after-refmac 1)
```

which will swap the colours of then new and old refmac map so that the post-refmac map has the same colour as the pre-refmac map and the pre-refmac map is coloured with a different colour.

<sup>35</sup>including sulfate or phosphate ions (in such a case, it is probably useful to do a "Rigid Body Fit Zone" on that new residue).

<sup>36</sup>For example, after a ligand search has been performed.

<sup>37</sup>*i.e.* Coot is idle and ignores all input.

<sup>38</sup>This is not an idea feature, of course and will be addressed in future.... Digressive Musing: If only computers were fast enough to run Refmac interactively...

## 5.19 Clear Pending Picks

Sometimes one can click on a button<sup>39</sup> unintentionally. This button is there for such a case. It clears the expectation of an atom pick. This works not only for modelling functions, but also geometry functions (such as Distance and Angle).

## 5.20 Delete

Single atoms or residues can be deleted from the molecule using “Delete...” from the “Model/Fit/Refine” dialog. Pressing this button results in a new dialog, with the options of “Residue” (the default), “Atom” and “Hydrogen Atoms”. Now click on an atom in the graphics - the deleted object will be the whole residue of the atom if “Residue” was selected and just that atom if “Atom” was selected.

If you want to delete multiple items you can either use check the “Keep Delete Active” check-button on this dialog or use the Ctrl key as you click on an atom. Either of these will keep the dialog open, ready for deletion of next item.

## 5.21 Sequence Assignment

You can assign a (fasta format) sequence to a molecule using:

```
(assign-fasta-sequence imol chain-id fasta-seq)
```

This function has been provided as a precursor to functions that will (as automatically as possible) mutate your current coordinates to one that has the desired sequence. It will be used in automatic side-chain assignment (at some stage in the future).

## 5.22 Building Links and Loops

Coot can make an attempt to build missing linking regions or loops<sup>40</sup>. This is an area of Coot that needs to be improved, currently O does it much better. We will have several different loop tools here<sup>41</sup>. For now:

```
(fit-gap imol chain-id start-resno stop-resno)
```

and

```
(fit-gap imol chain-id start-resno stop-resno sequence)
```

the second form will also mutate and try to rotamer fit the provided sequence.

Example usage: let's say for molecule number 0 in chain "A" we have residues up to 56 and then a gap after which we have residues 62 and beyond:

```
(fit-gap 0 "A" 57 61 "TYPWS")
```

## 5.23 Setting Occupancies

As well as the editing “Residue Info” to change occupancies of individual atoms, one can use a scripting function to change occupancies of a whole residue range:

```
(zero-occupancy-residue-range imol chain-id  
resno-start resno-last)
```

<sup>39</sup>such that Coot would subsequently expect an atom selection “pick” in the graphics window.

<sup>40</sup>the current single function doesn't always perform very well in tests, which is why it is currently available only in the scripting format.

<sup>41</sup>I suspect that there is not one tool that fits for all.

example usage:

```
(zero-occupancy-residue-range 0 "A" 23 28)
```

This is often useful to zero out a questionable loop before submitting for refinement. After refinement (with `refmac`) there should be relatively unbiased density in the resulting 2Fo-Fc-style and difference maps.

Similarly there is a function to reverse this operation:

```
(fill-occupancy-residue-range imol chain-id  
resno-start resno-last)
```

## Chapter 6

# Map-Related Features

### 6.1 Maps in General

Maps are “infinite,” not limited to pre-calculated volume (the “Everywhere You Click - There Is Electron Density” (EYC-TIED) paradigm) symmetry-related electron density is generated automatically. Maps are easily re-contoured. Simply use the scroll wheel on you mouse to alter the contour level (or -/+ on the keyboard).

Maps follow the molecule. As you recentre or move about the crystal, the map quickly follows. If your computer is not up to re-contouring all the maps for every frame, then use Draw → Dragged Map... to turn off this feature.

Unfortunately, there is a bug in map-reading. If the map is not a bona-fide CCP4 map<sup>1</sup>, then coot will crash. Sorry. A fix is in the works but “it’s complicated”.

### 6.2 Create a map

From MTZ, mmCIF and .phs (PHASES format) data use File → Read Dataset... From a CCP4 map use File → Read Map. After being generated/read, the map is immediately contoured and centred on the current rotation centre.

#### 6.2.1 Reading CIF data

There are several maps that can be generated from CIF files that contain observed  $F_s$ , calculated  $F_s$  and calculated phases:

`(read-cif-data-with-phases-fo-alpha-calc cif-file-name)` Calculate an atom map using  $F_{obs}$  and  $\alpha_{calc}$

`(read-cif-data-with-phases-2fo-fc cif-file-name)` Calculate an atom map using  $F_{obs}$ ,  $F_{calc}$  and  $\alpha_{calc}$

`(read-cif-data-with-phases-fo-fc cif-file-name)` Calculate an difference map using  $F_{obs}$ ,  $F_{calc}$  and  $\alpha_{calc}$ .

### 6.3 Map Contouring

Maps can be re-contoured using the middle-mouse scroll-wheel (buttons 4 and 5 in X Window System™ terminology). Scrolling the mouse wheel will change the map contour level and the map is redrawn. If you have several maps displayed then

---

<sup>1</sup>e.g. it’s a directory or a coordinate filename.

the map that is has its contour level changed can be set using **HID** → **Scrollwheel** → **Attach scroll-wheel to which map?**. If there is only one map displayed, then that is the map that has its contour level changed (no matter what the scroll-wheel is attached to in the menu). The level of the electron density is displayed in the top right hand corner of the OpenGL canvas.

Use **Keyboard** + or - to change the contour level if you don't have a scroll-wheel<sup>2</sup>.

If you are creating your map from an MTZ file, you can choose to click on the "is difference map" button on the Column Label selection widget (after a data set filename has been selected) then this map will be displayed in 2 colours corresponding to + and - the map contour level.

If you read in a map it is a difference map then there is a checkbox to tell Coot that.

If you want to tell Coot that a map is a difference map after it has been read, use:

```
(set-map-is-difference-map imol)
```

where *imol* is the molecule number.

By default the map radius<sup>3</sup> is 10Å. The default increment to the electron density depends on whether or not this is a difference map (0.05 e<sup>-</sup>/Å<sup>3</sup> for a "2Fo-Fc" style map and 0.005 e<sup>-</sup>/Å<sup>3</sup> for a difference map). You can change these using **Edit** → **Map Parameters** or by using the "Properties" button of a particular map in the Display Control (Display Manager) window.

## 6.4 Map contour "scrolling" limits

Usually one doesn't want to look at negative contour levels of a map<sup>4</sup>, so Coot has by default a limit that stops the contour level going beyond (less than) 0. To remove the limit:

```
(set-stop-scroll-iso-map 0) for a 2Fo-Fc style map
(set-stop-scroll-diff-map 0) for a difference map
```

To set the limits to negative (e.g. -0.6) levels:

```
(set-stop-scroll-iso-map-level -0.6)
```

and similarly:

```
(set-stop-scroll-diff-map-level -0.6)
```

where the level is specified in electrons/Å<sup>3</sup>.

## 6.5 Map Line Width

The width of the lines that describe the density can be changed like this:

```
(set-map-line-width 2)
```

The default line width is 1.

## 6.6 "Dynamic" Map colouring

By default, maps get coloured according to their molecule number. The starting colour (*i.e.* for molecule 0) is blue. The colour of a map can be changed by **Edit** →

---

<sup>2</sup>like I don't on my Mac.

<sup>3</sup>actually, it's a box.

<sup>4</sup>in a coot difference map you will get to see the negative level contoured at the inverted level of the positive level, what I mean is that you don't want to see the "positive" level going less than 0.



Map Colour... The map colour gets updated as you change the value in the colour selector<sup>5</sup>. Use “OK” to fix that colour.

## 6.7 Difference Map Colouring

For some strange reason, some crystallographers<sup>6</sup> like to have their difference maps coloured with red as positive and green as negative, this option is for them:

```
(set-swap-difference-map-colours 1)
```

## 6.8 Map Sampling

By default, the Shannon sampling factor is the conventional 1.5. Use larger values (Edit → Map Parameters → Sampling Rate) for smoother maps<sup>7</sup>.

## 6.9 Dragged Map

By default, the map is re-contoured at every frame during a drag (Ctrl Left-mouse). Sometimes this can be annoyingly slow and jerky so it is possible to turn it off: Draw → Dragged Map → No.

To change this by scripting:

```
(set-active-map-drag-flag 0)
```

## 6.10 Dynamic Map Sampling and Display Size

If activated (Edit → Map Parameters → Dynamic Map Sampling) the map will be re-sampled on a courser grid when the view is zoomed out. If “Display Size” is also activated, the box of electron density will be increased in size also. In this way, you can see electron density for big maps (many unit cells) and the graphics still remain rotatable.

## 6.11 Skeletonization

The skeleton (also known as “Bones”<sup>8</sup>) can be displayed for any map. A map can be skeletonized using Calculate → Map Skeleton... Use the option menu to choose the map and click “On” then “OK” to generate the map (the skeleton is off by default).

The level of the skeleton can be changed by using Edit → Skeleton Parameters... → Skeletonization Level... and corresponds to the electron density level in the map. By default this value is 1.2 map standard deviations. The amount of map can be changed using Edit → Skeleton Parameters... → Skeleton Box Radius...<sup>9</sup>. The units are in Ångströms, with 40 as the default value.

The skeleton is often recalculated as the screen centre changes - but not always since it can be an irritatingly slow calculation. If you want to force a regeneration of the displayed skeleton, simply centre on an atom (using the middle mouse button) or press the “s” key.

<sup>5</sup>takes you right back to the good old Frodo days, no?

<sup>6</sup>Jan Dohnalek, for instance.

<sup>7</sup>a value of 2.5 is often sufficient.

<sup>8</sup>If you’re living in Sweden... or Captain Kirk, that is.

<sup>9</sup>you may think it strange that a box has a radius, this is an idiosyncrasy of coot.

## 6.12 Masks

A map can be masked by a set of coordinates. Use the scripting function:

```
(mask-map-by-protein map-number coords-number 0)10.
```

This will create a new map that has density where there are no (close) coordinates. So for example, if you wanted to show the density around your ligand, you would create a coordinates file that contained all the protein except for the ligand and use those coordinates to mask the map.

There is no GUI interface to this feature at the moment.

### Example

If one wanted to show just the density around a ligand:

1. Make a pdb file that contains just the ligand and read it in to Coot - let's say it is molecule 1 and the ligand is residue 3 of chain "L".
2. Get a map that covers the ligand (e.g. from *refmac*). Let's say this map is molecule number 2.
3. Mask the map:

```
(mask-map-by-molecule 2 1 #f)
```

This creates a new map. Turn the other maps off, leaving only the masked map.

To get a nice rendered image, press F8 (see Section 3.4.2).

## 6.13 Trimming

If you want to remove all the atoms<sup>11</sup> that lie "outside the map" (i.e. in low density) you can use

```
(trim-molecule-by-map imol-coords imol-map density-level  
delete/zero-occ?)
```

where *delete/zero-occ?* is 0 to remove the atoms and 1 to set their occupancy to zero.

There is no GUI interface to this feature.

<sup>10</sup>the 0 is a placeholder for an as yet unimplemented feature (*invert?*).

<sup>11</sup>or set their occupancy to zero

## Chapter 7

# Validation

The validation functions are in the process of being written. In future there will be more functions, particularly those that will interface to other programs<sup>1</sup>.

### 7.1 Ramachandran Plots

Ramachandran plots are “dynamic”. When you change the molecule (*i.e.* move the coordinates of some of atoms) the Ramachandran plot gets updated to reflect those changes. Also the underlying  $\phi/\psi$  probability density changes according to the selected residue type (*i.e.* the residue under the mouse in the plot). There are 3 different residue types: GLY, PRO, and not-GLY-or-PRO<sup>2</sup>.

When you mouse over a representation of a residue (a little square or triangle<sup>3</sup>) the residue label pops up. The residue is “active” *i.e.* it can be clicked. The “graphics” view changes so that the C $\alpha$  of the selected residue is centred. In the Ramachandran plot window, the current residue is highlighted by a green square.

### 7.2 Chiral Volumes

The dictionary is used to identify the chiral atoms of each of the model’s residues. A clickable list is created of atoms whose chiral volume in the model is of a different sign to that in the dictionary.

### 7.3 Blobs: a.k.a. Unmodelled density

This is an interface to the Blobs dialog. A map and a set of coordinates that model the protein are required.

A blob is region of relatively high residual electron density that cannot be explained by a simple water. So, for example, sulfates, ligands, mis-placed sidechains or unbuilt terminal residues might appear as blobs. The blobs are in order, the biggest <sup>4</sup> at the top.

---

<sup>1</sup>such as the Richardsons’ reduce and probe

<sup>2</sup>the not-GLY-or-PRO is the most familiar Ramachandran plot.

<sup>3</sup>prolines have a grey outline rather than a black one, triangles are glycines.

<sup>4</sup>and therefore most interesting

## 7.4 Check Waters by Difference Map

Sometimes waters can be misplaced - taking the place of sidechains or ligands or crystallization agents such as phosphate for example<sup>5</sup>. In such cases the variance of the difference map can be used to identify them.

This function is also useful to check anomalous maps. Often waters are placed in density that is really a cation. If such an atom diffracts anomalously this can be identified and corrected.

By default the waters with a map variance greater than  $3.5\sigma$  are listed. One can be more rigorous by using a lower cut-off:

```
(set-check-waters-by-difference-map-sigma-level 3.0)
```

## 7.5 Validation Graphs

Coot provides several graphs that are useful for model validation (on a residue by residue basis): residue density fit, geometry distortion, temperature factor variance, peptide distortion and rotamer analysis.

### 7.5.1 Residue Density Fit

The residue density fit is by default scaled to a map that is calculated on the absolute scale. Some users use maps that have maps with density levels considerably different to this, which makes the residue density fit graph less useful. To correct for this you can use the scripting function:

```
(set-residue-density-fit-scale-factor factor)
```

where *factor* would be  $1/(4\sigma_{\text{map}})$  (as a rule of thumb).

(`residue-density-fit-scale-factor`) returns the current scale factor (default 1.0).

### 7.5.2 Rotamer Analysis

Residue rotamers are scored according to the prior likelihood. Note that when CD1 and CD2 of a PHE residue are exchanged (simply a nomenclature error) this can lead to large red blocks in the graph (apparently due to very unlikely rotamers). There are several other residues that can have nomenclature errors like this.

### 7.5.3 Temperature Factor Variance

### 7.5.4 Peptide $\omega$ Distortion

### 7.5.5 Geometry

---

<sup>5</sup>or the water should be more properly modelled as anisotropic or a split partial site

## Chapter 8

# Hints

### 8.1 Getting out of “Translate” Mode

If you get stuck in “translate” mode in the GL canvas (*i.e.* mouse does not rotate the view as you would expect) simply press and release the Ctrl key to return to “rotate” mode.

### 8.2 Getting out of “Label Atom Only” Mode

Similarly, if you are stuck in a mode where the “Model/Fit/Refine” buttons don’t work (the atoms are not selected, only the atom gets labelled), press and release the Shift key.

### 8.3 Button Labels

Button labels ending in “...” mean that a new dialog will pop-up when this button is pressed.

### 8.4 Picking

Note that left-mouse in the graphics window is used for both atom picking and rotating the view, so try not to click over an atom when trying to rotate the view when in atom selection mode.

### 8.5 Resizing View

Click and drag using right-mouse (up and down or left and right) to zoom in and out.

### 8.6 Map

If the “Display” button for the map in the “Display Manager” window stops working, close the “Display Control” window and re-open it. The button should now respond to clicks.

To change the map to which the scroll-wheel is attached, use HID → Scrollwheel → Attach Scrollwheel to which map?

## 8.7 Slow Computer Configuration

Several of the parameters of Coot are chosen because they are reasonable on my “middle-ground” development machine. However, these parameters can be tweaked so that slower computers perform better:

```
(set-smooth-scroll-steps 4) ; default 8
(set-smooth-scroll-limit 30) ; Angstroms
(set-residue-selection-flash-frames-number 3);
(set-skeleton-box-size 20.0) ; A (default 40).
(set-active-map-drag-flag 0) ; turn off recontouring every step
(set-idle-function-rotate-angle 1.5) ; turn up to 1.5 degrees
```

# Index

- `.coot.py`, 15
- `.coot`, 16
- `set-matrix`, 27
  
- anisotropic atoms, 23
- atom colouring, 14, 22
- atom info, 21
- atom label, brief, 21
- atom labeling, 21
- atom picking, 37
- auto-fit rotamer, 31
  
- background colour, 18
- backups, 16
- baton build, 29
- baton mode, 29
- big maps, 41
- blobs, 43
- bond thickness, 22
- bones, 41
  
- C $\alpha$  representation, 15
- C $\alpha$  symmetry representation, 24
- C-terminus, 35
- change contour level, 39
- cif dictionary, mmCIF dictionary, 28
- clashing residues, 31
- Clear Pending Picks, 37
- clipping, 18
- colour by chain, 14
- colouring, atoms, 22
- colouring, map, 40
- command line arguments, 8
- contouring, map, 12, 39
- coordinates format, 21
- crash, 9
- crash recovery, 9
- crosshairs, 18
  
- debugger, 9
- default refmac version, 36
- delete, 37
- density line thickness, 40
- depth-cueing, 14
- dictionary, cif, 8, 34
- difference map, 40
- difference map colours, 40
- Display Manager, 14
- dual conformations, 32
- Dunbrack, Roland, 30
  
- edit  $\chi$  angles, 32
- edit B-factors, 21
- edit occupancy, 21
- executing commands, 15
- EYC-TIED, 39
  
- file-name filtering, 15
- flip peptide, 32
- frame rate, 18
  
- gdb, 9
- goose, 23
- guile, 16
  
- hydrogens, 22
  
- keyboard rotation, 11
  
- ligand torsion angles, 32
- ligands, 33
- ligands, flexible, 34
  
- mainchain, 30
- map line width, 40
- masks, 42
- mean B-factor, 24
- median B-factor, 24
- merge molecules, 36
- molecule centre, 25
- molecule number, 13
- mouse, 8
- mouse buttons, 11
- multi-mutate, 33
- multiple coordinates files, 21
- mutate, 32
  
- NCS, 22
- NCS averaging, 23
- negative contour levels, 40
  
- OCA, 23

- OpenGL, 13
- origin marker, 14
- output, 19
- OXT atom, 35
  
- packing diagram, 24
- pepflip, 32
- phases format, 39
- picking, 45
- pink pointer, 18
- planes, 27
- python, 15
  
- quanta-buttons, 12
  
- Ramachandran plot, 43
- Rasmol, 7
- Raster3D, 14
- reading multiple pdb files, 21
- recentring view, 17
- recover session, 9
- redo, 17
- reference structures, 8
- reference manual, 7
- refine single click, 27
- refinement, 27
- refinement, rigid body, 28
- refmac, 36
- refmac map colour, 36
- refmac parameters, 36
- refmac, default, 36
- regularization, 27
- render, 14
- renumbering residues, 33
- residue info, 21
- resizing view, 45
- restore after crash, 17
- rigid body fit, 28
- rotamers, 30
- rotate/translate, manual, 28
- rotation centre pointer, 18
- running refmac, 36
  
- scheme, 16
- screen-z rotation, 12
- scripting, 15
- sequence view, 24
- single click refine, 27
- skeleton regeneration, 41
- skeleton, missing, 30
- skeletonization, 41
- slab, 18
- sliding, 17
- slow computer, 46
  
- startup dialog (state), 16
- startup settings (python), 15
- startup settings (scheme), 16
- state, 16
- symmetry, 23
  
- terminal oxygen, 35
- terminal residue, 35
- thickness of density lines, 40
- tooltips, 13
- torsion restraints, 27
- torsion angles, ligand, 32
- torsions, 32
- trackball, virtual, 12
- translate molecule, 25
- translation, keyboard, 12
- trimming atoms, 42
- tutorial, 7
  
- undo, 16
- unexplained density, 43
- unit cell, 18
- unmodelled density, 43
  
- version number, 13
- view matrix, 17
  
- waters, finding, 34
- web page, 9
- width, bonds, 22
  
- yellow box, 14
  
- z-rotation, 12
- zero occupancy, 24
- zoom, 11, 45
- zoom, slider, 12